Amendments to the Claims

This listing of claims replaces all prior versions and prior listings of claims in the application.

Listing of Claims

1. (original) An automated system for generating multiple solutions in near real time for n open pairings (where n is an integer) occurring in airline operations, which comprises:

a memory system having stored therein first memory objects defining flight and crew data, legality rules and other data relating to pairings of an airline;

an optimization server in electrical communication with said memory system and receiving user requests for updating said memory system and for generating solutions to cure said n open pairings; and

a microprocessor in electrical communication with said memory system and said optimization server for identifying all deadhead paths occurring among said n open pairings, generating solution components based upon said deadhead paths, storing said deadhead paths and said solution components in said memory system as second memory objects, and upon receiving from said optimization server a user request for curing said n open pairings, extracting said first memory objects and said second memory objects to form an open pairing model, formulating said open pairing model into an integer program data structure to simplify said open

pairing model for solution, and generating said multiple solutions based upon said integer program data structure to cure said n open pairings. '

2. (original) The automated system of claim 1, wherein said solution components include partial self-fixes, complete self-fixes, and one-way fixes.

3. (original) The automated system of claim 1, wherein said multiple solutions include one or more n-way swaps.

4. (original) The automated system of claim 1, wherein one of said solution components is a one-way fix in which an open pairing is cured by taking one or more flights from a single

remaining one of said n open pairings.

5. (currently amended) The automated system of claim 1, wherein said integer program data structure is provided by an integer program represented by:

$$\sum_{jt} x_{ij}^{t} = 1 \forall i, \quad \text{(i)}$$

$$\sum_{i,j,t} x_{ij}^{t}(f) + y_f - S_f = 1 \forall f, \text{ and} \quad \text{(ii)}$$

$$\text{Objective} = \min. \sum_{i,j}^{t} C_{i,j}^{t} x_{i,j}^{t} + \sum_f C_f y_f + \sum_f B_f S_f \quad \text{(iii)}$$

where "i" represents a first Pairing in a fix, whether broken or unbroken;

where "j" represents a second Pairing in a fix, whether broken or unbroken;

where "f" is an integer indicating a flight identifier;

where "$C_{i,j}^{t}$" and "$x_{i,j}^{t}$" represent a Cost to an airline in implementing a One-Way Fix "$x_{i,j}^{t}$" when $i \neq j$ and a Self Fix "$x_{i,j}^{t}$" when $i = j$;

where "C_f" represents a Cost to an airline for flight f which is not covered by any fixes;

where "y_f" is an integer, where $y_f = 0$ if a flight f is covered by a fix "x" and $y_f = 1$ if flight f is not covered by a fix "x";

where "B_f" represents a Cost of crew over-coverage for flight f;

where "S_f" represents a slack factor referring to a over-coverage of crew to guarantee a feasibility of a solution; and

where "min." represents a minimum function, in which a smallest number in a data set is returned.

6. (original) The automated system of claim 1, wherein said integer program data structure is comprised of both a crew constraint set and a flight constraint set.

7. (original) The automated system of claim 1, wherein said solution components are one-way swaps and two-way swaps, and a matching transformation rather than said integer program data structure is used to generate said multiple solutions.

8. (original) A method of generating in near real time multiple solutions for n open pairings (where n is an integer) in a microprocessor which is in electrical communication with a memory system and an optimization server, which comprises the steps of:

receiving memory objects from said memory system which include all crew and flight records, and legality rules defining operations affecting said n open pairings;

identifying all deadhead paths occurring among said n open pairings;

generating solution components based upon said deadhead paths for reassigning flights among said n open pairings, and storing said solution components in said memory system;

upon receiving a request from a user by way of said optimization server to generate solutions to cure an open pairing, extracting one-way swaps and two-way swaps from said solution components to form an open pairing problem; and

performing a matching transformation on said open pairing problem based upon said one-way swaps and said two-way swaps to provide said multiple solutions.

9. (original) The method of claim 8 further including the step of selecting from said multiple solutions a solution of least cost for transmission to said user by way of said optimization server.

10. (original) The method of claim 8, wherein said matching transformation is comprised of the following steps:

for each original vertex and mirror vertex comprising a pair representing an open pairing, connecting said original vertex and said mirror vertex with an arc;

for any two of said n open pairings, select one of said one-way swaps and said two-way swaps between said any two of said n open pairings having a least cost;

joining original vertices of said any two of said n open pairings with a second arc having said least cost; and

for each mirror vertex pair, join mirror vertices of said mirror vertex pair with an arc of cost zero.

11. (original) A method of generating in near real time multiple solutions for n open pairings (where n is an integer) in a microprocessor in electrical communication with a memory system and an optimization server, which comprises:

receiving memory objects from said memory system which include crew and flight data, and legality rules relating to said n open pairings;

generating solution components for reassigning flights among said n open pairings, and storing said solution components in said memory system;

upon receiving a request from a user by way of said optimization server to generate solutions to cure an open pairing, extracting complete self-fixes, partial self-fixes, and one-way fixes from said solution components to form an open pairing problem;

applying an integer program to said open pairing problem to formulate said pairing problem into an integer program data structure; and

generating said multiple solutions for said n open pairings based upon said integer program data structure.

12. (cancelled)

13. (original) The method of claim 11, wherein said integer program includes a crew constraint set and a flight constraint set.

14. (original) The method of claim 11, wherein said multiple solutions includes one or more n-way swaps.

15. (currently amended) The method of claim 11, wherein said integer program

is represented by:

$$\sum_{jt} x_{ij}^{t} = 1 \forall i, \qquad \text{(i)}$$

$$\sum_{i,j,t} x_{ij}^{t}(f) + y_f - S_f = 1 \forall f, \text{ and} \quad \text{(ii)}$$

$$\text{Objective} = \min. \sum_{i,j}^{t} C_{i,j}^{t} x_{i,j}^{t} + \sum_f C_f y_f + \sum_f B_f S_f \quad \text{(iii)}$$

where "i" represents a first Pairing, whether broken or unbroken;

where "j" represents a second Pairing in a fix, whether broken or unbroken;

where "f" is an integer indicating a flight identifier;

where "$C^t_{i,j}$" and "$x^t_{i,j}$" represent a Cost to an airline in implementing a One-Way Fix "$x^t_{i,j}$" when $i \neq j$ and a Self Fix "$x^t_{i,j}$" when $i = j$;

where "C_f" represents a Cost to an airline for flight f which is not covered by any fixes;

where "y_f" is an integer, where $y_f = 0$ if a flight f is covered by a fix "x" and $y_f = 1$ if flight f is not covered by a fix "x";

where "B_f" represents a Cost of crew over-coverage for flight f;

where "S_f" represents a slack factor referring to a over-coverage of crew to guarantee a feasibility of a solution; and

where "min." represents a minimum function, in which a smallest number in a data set is returned.

16. (original) The method of claim 11, wherein said solution components include deadhead paths.